SLAC-TN-07-029

# Fast Track Finding in the ILC's Silicon Detector, SiD01

David E. Baker

Office of Science, Science Undergraduate Laboratory Internship Program

Carnegie Mellon University

Stanford Linear Accelerator Center

Stanford, CA

August 24, 2007

Prepared in partial fulfillment of the requirement of the Office of Science, Department of Energy's Science Undergraduate Laboratory Internship under the direction of Norman Graf in the Linear Collider Detector Department at the Stanford Linear Accelerator Center.

Participant: _____

Signature

Research Advisor: _____

Signature

Work supported in part by US Department of Energy contract DE-AC02-76SF00515

**Abstract**

A fast track finder is presented which, unlike its more efficient, more computationally costly $O(n^3)$ time counterparts, tracks particles in $O(n)$ time (for $n$ being the number of hits). Developed as a tool for processing data from the ILC's proposed SiD detector, development of this fast track finder began with that proposed by Pablo Yepes in 1996 [1] and adjusted to accommodate the changes in geometry of the SiD detector. First, space within the detector is voxellated, with hits assigned to voxels according to their $r$, $\varphi$, and $\eta$ coordinates. A hit on the outermost layer is selected, and a "sample space" is built from the hits in the selected hit's surrounding voxels. The hit in the sample space with the smallest distance to the first is then selected, and the sample space recalculated for this hit. This process continues until the list of hits becomes large enough, at which point the helical circle in the $x$, $y$ plane is conformally mapped to a line in the $x$', $y$' plane, and hits are chosen from the sample spaces of the previous fit by selecting the hits which fit a line to the previously selected points with the smallest $\chi^2$. Track finding terminates when the innermost layer has been reached or no hit in the sample space fits those previously selected to an acceptable $\chi^2$. Again, a hit on the outermost layer is selected and the process repeats until no assignable hits remain. The algorithm proved to be very efficient on artificial diagnostic events, such as one hundred muons scattered at momenta of 1 GeV/c to 10 GeV/c. Unfortunately, when tracking simulated events corresponding to actual physics, the track finder's efficiency decreased drastically (mostly due to signal noise), though future data cleaning programs could noticeably increase its efficiency on these events.

# Table of Contents

**Introduction**

After the BaBar experiment, the last particle physics experiment at SLAC, has been completed, particle physicists will need a new tool to study high-energy physics and lepton interactions. The proposed tool is the International Linear Collider (ILC), to be 40 km long and collide positron/electron ($e^+/e^-$) pairs at energies of up to 1 TeV. This device is expected to implement innovative methods of $e^+/e^-$ storage, beam acceleration, and particle detection. One proposed detector is the "SiD" model, which replaces layers of drift chambers or scintillating material with silicon panels measured on the micrometer scale. These silicon panels would be able to resolve the location of a particle hit on its surface to five micrometers, resulting in a highly improved digital tracking mechanism.

The SiD model consists of five finely-faceted vertex detector layers surrounding the immediate vicinity of collisions, followed by some number of tracker barrels: layers of silicon paneling outside the vertex detector meant to track particles moving through the collider's magnetic field. This study is intended to investigate algorithms which would identify corresponding particle hits through the vertex detector and tracker barrels, and reconstruct particle tracks. Secondary considerations of the study would include optimizing the accuracy of the tracking algorithm with respect to the number of tracker barrels and the cost of the device.

The "fast track finder" of Dr. Pablo Yepes [1] was used as a launching point for this study. It is already an efficient algorithm, and the one developed here implements the same methods: the space of the detector is voxellated, with hits in each voxel stored together in data structures. A hit on the outermost layer is then selected, and the closest hit on the second-outermost layer found (with the assumption that in the outermost layer, hit density is small enough that these two hits must be associated). Because charged particles in a magnetic field move in helices, and it is computationally easier to fit lines than circles, hit coordinates are conformally mapped into a "feature space," which maps circles in the real space into lines. A linear projection between the two points is then used to estimate the locations of associated points on inner layers. When enough points have been assigned to a track, cuts can be made on $\chi^2$ fits to lines in feature space and circles in real space to determine which tracks probably belong to particles. This process is repeated until all in-range hits have been assigned to tracks to a desirable level of accuracy.

Challenges in the adaptation of the tracker include tracking particles through the decreased number of tracker barrel layers (detectors outfitted with drift chambers commonly have many more layers of tracking material) as well as reconstructing tracks that decayed outside the vertex detector, such as the decay products of neutral hadrons. With computing power freed due to a decreased number of hits, more rigorous methods of predicting the location of corresponding hits can be integrated into the algorithm, resulting in an increase in accuracy without suffering a loss of efficiency.

Data for testing the algorithm was generated by running Monte Carlo simulations, designed to replicate the detector's geometry and magnetic field. Debugging was done using single-particle or few-particle events, the physics of which was not relevant to the detector, while efficiency studies used these events and actual simulations of the intended $e^+/e^-$ collisions at 500 GeV and 1 TeV. Tracks produced by the algorithm could be compared to the Monte Carlo data for the event, and the differences between the opening angle, distance of closest approach, and curvature, of the reconstructed tracks against the tracks generated by Monte Carlo tracks could be used to measure the algorithm's efficiency.

**Materials**

The Silicon Detector over which this tracker was implemented had two parts: a vertex detector and a tracker barrel. The vertex detector consists of silicon wafers with 20 μm pixel readout, arranged in five cylindrical layers concentric about the beam line at the interaction point. The tracker barrel consists of five more layers of similarly arranged silicon strips, spaced much more widely in radius than those in the tracker barrel. The dimensions of the vertex cylinders and tracker barrel cylinders are given in tables 1.a and 1.c, respectively. The vertex detector and tracker barrel are each outfitted with four disks lying perpendicular to the detectors' cylindrical axes. Dimensions of these disks are given in tables 1.b and 1.d, respectively. To track particles moving in the forward or backward direction, each disk in the vertex detector is covered with forward-facing silicon wafers, again with 20 μm pixel readout. Each disk in the tracker barrel is covered in two layers of elongated silicon sensor strips, set at an angle from one another so that different strips overlap.

Collisions in the vertex detector produce jets of particles flying outward through the detector's magnetic field. As a charged particle passes through a layer of silicon, a signal is read

into positron sensors on an attached chip which can resolve the position of the hit on the silicon plate to 5 μm [2]. This provides, for each event, a set of three-coordinate points, each of which represents the location where a charged particle crossed a silicon wafer. Particles then pass through the outer tracker barrels, which make more (albeit less-precise) measurements of particle location. In the outer tracker endcap disks, the intersection area of two activated strips provides the three dimensional space point of the particle which activated them.

**Procedure**

The sample data used consisted of Monte Carlo simulations, which matched the geometry of the detector and the energies specific to the ILC [3], which was read in to the program through a Java Analysis Studio (JAS3) driver. The space of the detector was then digitally divided into equally-sized voxels in the $r$, $\varphi$, $\eta$[1] coordinates (the size of which depended on hard-coded parameters within the program) and each voxel was assigned a list of the hits whose coordinates fell within its domain. The algorithm began by randomly selecting a hit on the outermost populated layer and building a 'sample space' about that point. The sample space consisted of all the hits which fell into voxels immediately about the selected point. For instance, if the selected point was within the $i^{th}$ layer, $j^{th}$ $\varphi$ and $k^{th}$ $\eta$ voxel, (hereafter labeled $r_i$, $\varphi_j$, $\eta_k$), the sample space would be filled with the hits assigned to all the voxels with the labeled by

$$\begin{array}{ccc} r_{i+n}, \varphi_{j-1}, \eta_{k-1} & r_{i+n}, \varphi_{j}, \eta_{k-1} & r_{i+n}, \varphi_{j+1}, \eta_{k-1} \\ r_{i+n}, \varphi_{j-1}, \eta_{k} & r_{i+n}, \varphi_{j}, \eta_{k} & r_{i+n}, \varphi_{j+1}, \eta_{k} \\ r_{i+n}, \varphi_{j-1}, \eta_{k+1} & r_{i+n}, \varphi_{j}, \eta_{k+1} & r_{i+n}, \varphi_{j+1}, \eta_{k+1} \end{array}$$

Here, $n$ ranges from $1 \leq n \leq N$, where $N$ was left as a parameter of the program. Since hit density on the outermost layers is significantly smaller than hit density on the innermost layers, the simplest way to connect a hit with the first was to select the most proximal hit in the sample space of the first. Here, the “most proximal hit” was the hit with the smallest distance $d$ to the hit previously selected, where for two hits $j$ and $k$, $d$ was defined by

$$d_{jk} = |r_j - r_k| \times (|\varphi_j - \varphi_k| + |\eta_j - \eta_k|)$$

The algorithm constructed tracks in this way until a certain number of hits had been assembled. (The exact number left as a parameter in the program, though since three coplanar hits are required to uniquely define a circle, there had to be at least three.)

1 Here, $r$ is the distance orthogonal to the beam line, $\varphi$ is the angle in the plane orthogonal the beam line, and $\eta$ is the pseudorapidity, $\eta = -\ln [\ \tan(\theta/2)\ ]$, which (at these energies) is well approximated by $\eta = -3z / (\ |z| + r\ )$.

Once the track was large enough, the hits in the track were to be fit to a circle in the $x$, $y$ plane, and points could be chosen from the sample space by choosing whichever best fits this circle. Since fitting points to a circle is computationally difficult, the following conformal mapping:

$$x' = \frac{x - x_0}{r^2}, \quad y' = \frac{y_0 - y}{r^2}, \quad \text{where} \quad r^2 = (x - x_0)^2 + (y - y_0)^2,$$

was implemented, which defines a one-to-one mapping from circles that lie on the point $(x_0, y_0)$ in real $(x, y)$ space to lines through the origin in feature $(x', y')$ space (see Figure 1). Once enough hits on the outermost layers were selected, each hit on the track and in the sample space was transformed, taking the $x$ and $y$ coordinates of the first hit as $x_0$ and $y_0$. The next hit was selected by iterating through the hits in the sample space of the previous hit and selecting the hit which, when appended to the list of hits already discovered, provided the line in feature space with the smallest $\chi^2$ value. This hit was then added to the track, and the sample space recalculated from its location. The process repeated until either the innermost layer had been reached or the sample space for the last hit found was empty. If the set of hits collected met certain criteria[2], the hits in the track were permanently removed from the sample space, and the purity and fitting properties of the track evaluated.

**Analysis & Results**

To measure the efficiency of the detector, hits in the Monte Carlo simulations which corresponded to the same particles were assembled as tracks. Both the found tracks and the Monte Carlo tracks were fit to helices as circles in the $x, y$ plane and lines in the $s, z$ plane (where $s$ is the arc-length about the helix), and each such helix was uniquely defined by five parameters:

- $d_0$, the distance of closest approach of the closest hit to the circle of the helix in the $x$, $y$ plane
- $\varphi_0$, the azimuthal angle of the momentum of the particle at the point of closest approach
- $\Omega$, or $1/R$, where $R$ is the radius of curvature of the circle in the $x, y$ plane
- $z_0$, the $z$ position of the track at the distance of closest approach from some reference point
- $\tan\lambda$ is the slope, $ds/dz$, of the helix in the $s$, $z$ plane, defined from the momentum $\vec{p}$ by

2 These criteria were variable and based on parameters hard-coded into the program. Often, cuts were made on the number of hits found, the $\chi^2$ values of linear fits in feature space, and circular fits in real space. In this analysis, we required nothing but the number of hits on the track be greater than seven.

$$\tan\lambda = \frac{p_z}{\sqrt{p_x^2 + p_y^2}}.$$

(For more information on helical track parameters, as well as how they are calculated, see [5].) Since the values of $d_0$ and $z_0$ could change drastically depending on which hit the finder concluded its search on, the analysis of this track finder focused on the parameters $\varphi_0$, $\Omega$, and $\lambda$.[3]

The track finder was tested on an artificial data set consisting of 1,000 events of 100 muons, scattered at momenta varying from 1 GeV/c to 10 GeV/c. Of the 1,000 events, all were processed in under one second (see Figure 2) with a mean processing time of 300 milliseconds. The 'efficiency' of the track finder at a given momentum is defined as the number of tracks found by the track finder in a given momentum range (subject to a purity cut[4]) divided by the number of findable tracks in that momentum range. The efficiency of the track finder is plotted for the 1,000 events of 100 scattered muons in Figure 3. As can be seen in the histogram, in the 1 GeV/c momentum range, the efficiency was greater than one, which indicates the discovery of many fake tracks. The track finder seems to be most efficient in the momentum range about 2.0 GeV/c, where the efficiency is nearly 1. This is curious, because we would expect higher momentum particles, the motion of which is more linear, to be tracked more easily than lower momentum particles, the motion of which is more helical.

Unfortunately, when run on Monte Carlo simulations of physically relevant events, these results could not be replicated. For instance, when run on 100 simulated $\tau^+/\tau^-$ collisions (with six resultant jets), though each event was processed in under 1 second, with an average processing time of 404 milliseconds, the efficiency of the tracker rarely exceeded 0.3 (see Figure 4).

**Conclusion**

The fast track finder presented here proved to track particles with considerable efficiency, operating at speeds hundreds, or even thousands, of times faster than track finders which work in $O(n^3)$ time. It will undoubtedly prove to be a very beneficial tool to have at hand for fast data processing and data mining after the ILC comes online.

Curiously, this track finder's performance differs from most others in that it shows greater efficiency at tracking low momentum particles, which move as helices, than high momentum

[3] In the author's opinion, $\Omega$ is the most important parameter, since it maps directly onto the particle's momentum.

[4] We define "purity" as the greatest number of hits on a track belonging to a single particle divided by the number of hits on the track. In this analysis, we required the purity of findable tracks to be greater than 0.75.

particles, the motion of which is much more linear. Since most other track finders track high momentum particles more efficiently, it would be interesting to study the cause of this track finder operating well in the low regime. If this effect could be replicated in other track finders,

Though this track finder performed well on clean sets of sample data, its performance was found to be lacking on events simulating actual physics. Background noise, such as ionized electrons from the silicon plates being registered as hits, low momentum particles, and daughter particles created outside the vertex tracker, confused the track finder. This interference caused the finder to track as many as four times the number of particles in the event. Since programs which will find and remove such interference are under development by members of the SiD Tracking team, we can conclude that further studies of this algorithms' efficiency should be conducted when it can be run in tandem with such data processing algorithms. The ability to track actual events with the speed and efficiency exhibited by the track finder on the artificial data could prove to be indispensible for processing data from modern detectors.

## References


[1] P. Yepes, *"A fast track pattern recognition,"* Nuclear Instruments & Methods in Physics Research A 380 (1996) pp. 582-585.

[2] Silicon Detector (v1) for the Linear Collider at the SLAC Confluence
http://confluence.slac.stanford.edu/display/ilc/sid01

[3] Anonymous FTP for the Linear Collider at the SLAC Confluence
ftp://ftp-lcd.slac.stanford.edu/lcd/ILC/

[4] T. Ferbel, *"Techniques and Concepts of High-Energy Physics,"* St. Croix (1988), pp. 407 – 499

[5] T. Krämer, *"Track Parameters in LCIO,"* LC-DET-2006-004 (2006)

| Vertex Layer | Radius (cm) | Length (cm) |
|---|---|---|
| 1 | 1.46 cm | ± 6.25 cm |
| 2 | 2.26 cm | ± 6.25 cm |
| 3 | 3.54 cm | ± 6.25 cm |
| 4 | 4.80 cm | ± 6.25 cm |
| 5 | 6.04 cm | ± 6.25 cm |

| Vertex Disk | Radius (cm) | Dist (z, cm) |
|---|---|---|
| 1 | 1.4 cm | ± 7.21 cm |
| 2 | 1.6 cm | ± 9.05 cm |
| 3 | 1.8 cm | ± 12.19 cm |
| 4 | 2.0 cm | ± 17.03 cm |

Table 1.a (above): radius and length along the cylindrical axis of the silicon layers in the vertex detector.

Table 1.b (below): the dimensions of the vertex endcap disks, where "dist" measures the extant distance from the assumed interaction point (at the origin). Each disk has an outer radius of 7.1 cm.

| Tracker Layer | Radius (cm) | Length (cm) |
|---|---|---|
| 1 | 21.8 cm | ± 55.8 cm |
| 2 | 46.8 cm | ± 82.5 cm |
| 3 | 71.8 cm | ± 108.3 cm |
| 4 | 96.8 cm | ± 134.7 cm |
| 5 | 121.8 cm | ± 160.6 cm |

| Tracker Disk | Radius (cm) | Dist (z, cm) |
|---|---|---|
| 1 | 49.4 cm | ± 85.5 cm |
| 2 | 74.7 cm | ± 111.4 cm |
| 3 | 99.9 cm | ± 137.8 cm |
| 4 | 125.0 cm | ± 163.6 cm |

Table 1.c (above): radius and length along the cylindrical axis of the barrel layers in the outer tracker.

Table 1.d (below): the dimensions of the tracker barrel endcap disks, as in Table 1.b. Each disk has an inner radius of 20.7 cm and a second layer of sensors 0.40 cm further from the origin than the given "dist".

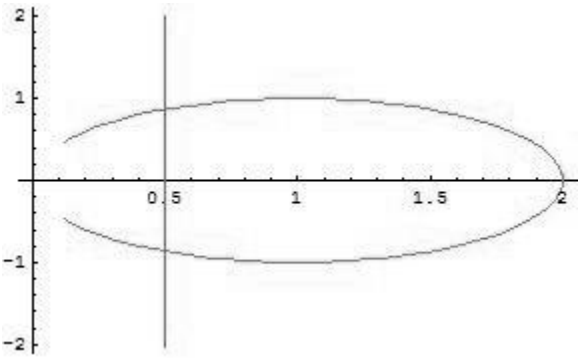


Figure 1: A parametric plot of the given conformal transformation, mapping the line $x = 1$, $y = t$ to the ellipse $x = 2\cos(t)$, $y = \sin(t)$.

For $w = x' + iy'$, $z = x + iy$, and $w(z) = z^{-1}$,

$$w = x' + iy' = \frac{1}{z} = \frac{1}{x + iy}.$$

Multiplying the numerator and denominator by the denominator's complex conjugate,

$$x' + iy' = \frac{1}{x + iy}\left(\frac{x - iy}{x - iy}\right) = \frac{x - iy}{x^2 + y^2}.$$

Matching real and imaginary parts, we have

$$x' = \frac{x}{\quad} \qquad y' = \frac{-y}{\quad}$$

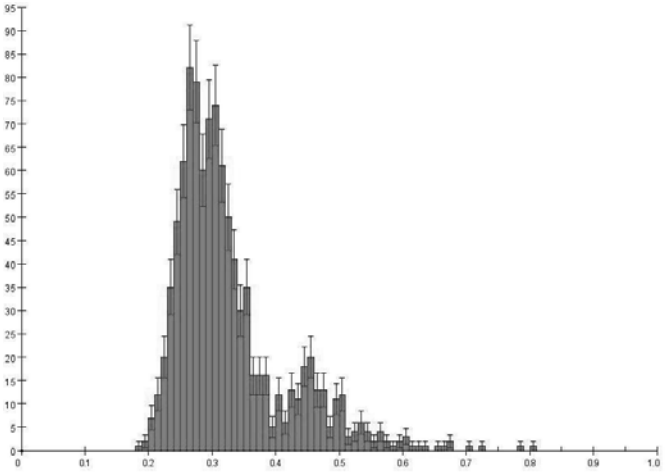

Figure 2: A histogram of the time taken to track each of the thousand artificial events, where the *y*-axis is measured in units of "seconds."

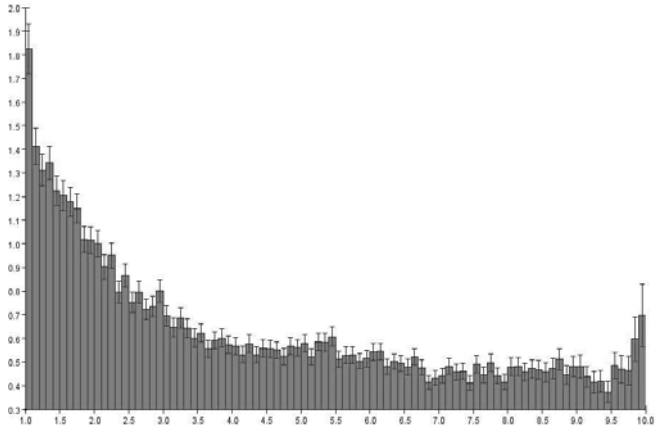

Figure 3: A histogram of the track finder's 'efficiency,' as previously defined, over the range (on the *y*-axis) of 1 GeV/c to 10 GeV/c.

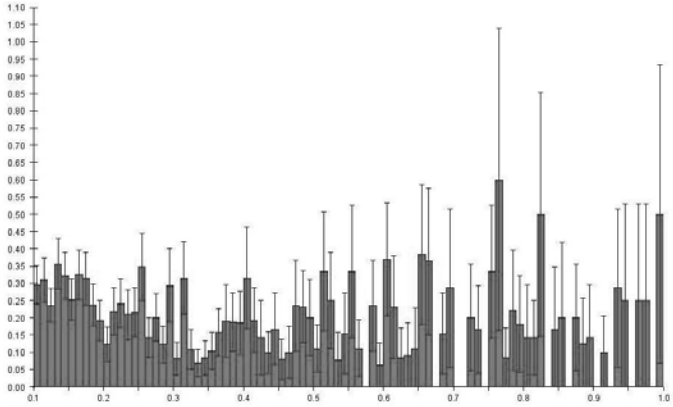

Figure 4: A histogram of the track finder's 'efficiency,' as previously defined, over 100 simulated $\tau^+/\tau^-$ collisions, each with six jets.